The Advent of Uncrackable Passwords[1], 3rd Edition

[1] Obviously no password is actually uncrackable, in this document the term "uncrackable" refers to "realistically or technically uncrackable." Please refer to the remainder of the document for more details.

The Importance of a New Solution

For years the world has been using passwords and passphrases as the sole means of security; but increasingly in today's high-tech world of sharks, conspiracies, and digital paranoia, there has been a heavy call for a new way to protect data and transactions over the web. Unfortunately, the previous solutions built around encryption of such data are no longer enough: at this moment no one knows who to trust, where to go, or what to do about it.

With lawsuits and whispers of scandal revolving around the major encryption standards of today (Zimmermann), internet service providers turning over all consumer data and transactions to the NSA ("AT&T sued over NSA eavesdropping", 2006) ("Whistleblower outs NSA's secret spy room at AT&T", 2006), laws passed allowing government eavesdropping on all citizens (Risen, 2005), and many more attacks against private individuals and corporations alike, by international governments, individuals, and other companies; passwords are definitely no longer enough.

At the same time, there are no viable alternatives to passwords. The only "alternates" are biometric security, ID cards, and security tokens; all of which are bulky and unfeasible, and just as vulnerable to attack in their pure data (binary) form (Password, 2006). It is obvious that passwords are the most convenient form of data protection, and as such are the most widely used. Passwords can be broken by brute force or dictionary attacks, both of which are made possible by the nature passwords/passphrases: simply an (un)ordered list of letters and/or symbols that can theoretically be cycled through and attempted until all possibilities are exhausted and/or the password is found (GeodSoft).

After realizing just how important a role passwords play in the lives of millions everyday, and how vulnerable such passwords are, yet how amazingly convenient their usage is, it becomes quite obvious that the ultimate solution would be an "uncrackable" password. However, such a password does not, and cannot exist. Not even in theory. But an "uncrackable" password is *not* needed; rather a *technically* uncrackable password is the key. Such a password does not need an infinite amount of time to be cracked, but rather take enough time and resources (ideally, years and thousands of dollars) for each password to be cracked, such that it becomes unfeasible for government agencies to crack such a password. And such a thing *is*, believe it or not, possible!

The Challenge

The problem with the modern password is the modern machine. Current home desktop machines can run through thousands of possible password combinations in a brute-force a minute. Realistically speaking, the government agencies use clustered machines many times costlier and hundreds if not thousands of times more powerful.

Given such a fact, the task at hand seems daunting. Using the standard base of characters, i.e. all possible character combinations from the keyboard,
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&()-_+=~`[]{}|\:;"'<>,.?/*
There are only a total of 68 characters. This does *not* include lower and uppercase combinations, seeing

as the current generation of operating systems store password hashes in a standard case-less format[2]. It takes close to no time at all for clustered machines at a business facility to crack such a password, and takes government computers even less time. It becomes apparent that the solution should come in the form of a way that complicate matters, introducing more variables into the mix, and complicating the results in such a way that would take many times more resources to find the password.

The Solution

With the standard 68 character set cited above, it is possible to calculate the number of possible permutations that a password or passphrase can take. As also mentioned above, 99%+ of the systems today use the LM security hash, which caps passwords at 14 characters in length and a single upper/lower case form. Since the password is split into 2 halves, and both are compared against a single hash table, in effect the number of characters is reduced to only 7 characters.

Total Permutations[3]: $68 + (68)^2 + (68)^3 + (68)^4 + (68)^5 + (68)^6 + (68)^7 = 6.823331935 \times 10^{12}$

While that may seem like an overly-large number, and looks to take an enormous amount of time to guess; given the processing power of today's machines, the availability of "Rainbow Tables," and the sheer resource strength of the various governments, it really is not altogether that high. While it *does* offer a significant protection against an average hacker, and it may tax a single home machine for one or two days without the use of rainbow tables, it is nowhere near complicated enough to stall government machines.

Enter the *other* 1000 (and then some) characters! Yes, there are indeed, hundreds more non-printable Unicode characters. These include commonly used symbols like ¶ and ↕, accented characters like ă and Ā, nonsense symbols like þ or even characters that *look* normal but aren't such as 0 and W. A password employing these characters can defeat *any* attempt at being cracked so long as the cracker has no idea that the "crackee" is using such characters in his or her password. At the moment, not a single password cracking program uses these characters in its dictionary making it *impossible* to crack!

All though there is no way to positively know, the government also does not use these cracking techniques, here is why: Using the same model employed above to discover the total number of

[2] Password crackers operate by taking the two halves of the (NT)LM passwords: the first 7 letters, and the second 7 letters. LM passwords cannot exceed this 14 character limit, and 99%+ of NTLM systems revert to LM upon query; which means that the additional security provided by NTLM is useless, and upper/lower case has no impact. The attack is then performed on both halves of the password simultaneously, thus reducing the number of characters from 14 to 7.

[3] 68 possible letters for each location, hence, 68 possible one-character passwords, 68X68 possible two-character passwords and so on and so forth until the seven-character combinations.

permutations, but with a different base, we can calculate the number of possible permutations to the base 700[4], giving us approximately $8.247211731 \times 10^{19}$!

This truly is a big number, and no matter which system does the cracking, this number will always take 12,086,781[5] times *longer* than the first password-type to crack! For example, if a password takes a home user 2 days or 48 hours to crack, it will now take 580,165,478 hours, or 67.15 thousand years! If the government could crack a password utilizing the full character set in an hour, it will now take them 1.4 thousand years to crack it! Even if they were to crack a password a minute, it would still take 22.3 *years* to crack a *single* password!

Implementation

To enter such a password, all you need is an alt key and a number pad. For example, to produce the character µ, simply hold down the right alt key, and on the number pad enter the key sequence 2-3-0, then let go of the alt button (this method only exists in Windows at the moment. Other operating systems may have their own methods of entering such characters). To enter the character Ğ simply use 3-2-1 instead!

While this may seem complicated, it really isn't. Many people memorize 20-character passphrases at the moment without needing to write them down. Instead of remembering eIEO86#44*&$uk; remembering 9-5-3 9-2-5 9-6-7 is *much* easier. Make use of logical/visual patterns on the keypad and it becomes easier still: 9-8-7 1-2-3 4-5-6 is just as strong!

Conclusion

Once a password, in the most favorable of conditions, takes years to crack on the most powerful of the government's clustered systems, it becomes what we call *virtually uncrackable.* At this rate, a government has to crack every single password it comes across to determine whether or not a document is of any importance. Even *if* we assume that only items of sensitive nature are protected with such a protection scheme, there remains the fact that in 22 years, a sensitive document becomes worthless. In 22 years, nations form, countries die, alliances break, and cities tumble. It 22 years, your boss will be dead, and that email you wrote criticizing him will be worthless.

The most important thing about this technology is, it is fully adaptable. Anytime a stronger method of encryption is born, this can be applied to it. You can apply this to simple text-ciphers, atbash encryption, PGP, or *any other encryption standard ever imaginable!* With it you can protect Windows or Linux, you can use it for good or for bad, and no one will ever know. Best of all, it is free and easy to use.

[4] Under the assumption that there are only 700 characters total (non-printable + printable). In reality there are at couple hundred more, but in order to keep things "believable" 700 will work just as well. The actual number is somewhere in the ballpark of 1200-1400 possible different characters!
[5] This is the second number of permutations divided by the first.

Whereas people were once memorizing 20 character passwords, now a 3 character password that takes 9 keystrokes to enter prevails.

Bibliography

AT&T sued over NSA eavesdropping. (2006, 1 31). Retrieved 4 10, 2006 from http://www.spamdailynews.com/: http://www.spamdailynews.com/publish/ATT_sued_over_NSA_eavesdropping.asp.

Electronic Frontier Foundation vs. AT&T Corporation (United States District Court, Northern District of California January 31, 2006).

Password. (2006). *Wikipedia*, 4.

Whistleblower outs NSA's secret spy room at AT&T. (2006, 4 8). Retrieved 4 10, 2006 from http://www.spamdailynews.com: http://www.spamdailynews.com/publish/ATT_tech_outs_NSA_spy_room.asp.

GeodSoft. (n.d.). *Password Cracking Goals, Techniques and Relative Merits and Cracking Times of Different Techniques.* Retrieved 4 10, 2006 from http://geodsoft.com/: http://geodsoft.com/howto/password/cracking_passwords.htm.

Risen, J. E. (2005, 12 16). *Bush Lets U.S. Spy on Callers Without Courts.* Retrieved 4 10, 2006 from http://www.nytimes.com/: http://www.nytimes.com/2005/12/16/politics/16program.html?ex=1292389200&en=e32072d786623ac1&ei=5090&partner=rssuserland&emc=rss.

Zimmermann, P. (n.d.). *PGP.* Retrieved 4 10, 2006 from http://www.philzimmermann.com/: http://www.philzimmermann.com/EN/faq/faq.html.

Zimmermann, P. (1996, February 2). Interview with author of PGP (Pretty Good Privacy). (R. Hoffman Interviewer)